基于蚁群算法的 SDN 数据中心网络大象流调度研究 *

李宏慧，杨　光，路海亮，付学良，申志军

(内蒙古农业大学 计算机与信息工程学院，呼和浩特 010010)

摘　要：针对传统方法调度大象流时容易造成数据中心网络拥塞和负载不均衡等问题，提出一种基于蚁群算法的 SDN（software defined network）数据中心网络流量调度算法 ACO-SDN。对大象流调度问题建立整型线性规划 ILP（integral linear programing）模型，优化目标为最小化最大链路利用率。通过重定义蚁群算法的参数和操作求解 ILP 模型，得到大象流重路由的最优路径。实验结果表明，与 ECMP(equal-cost multi-path routing）和 GFF（global first fit）流量调度算法相比，ACO-SDN 算法降低了网络最大链路利用率，有效地提高了网络对分带宽。

关键词：数据中心网络；软件定义网络；大象流调度；蚁群算法

中图分类号：TP393.07　　doi: 10.3969/j.issn.1001-3695.2018.07.0430

Flow scheduling of elephant flows in SDN data center network based on ant colony algorithm

Li Honghui, Yang Guang, Lu Hailiang, Fu Xueliang, Shen Zhijun

(*College of Computer & Information Engineering Inner Mongolia Agricultural University*, *Hohhot* 010018, *China*)

Abstract: The traditional methods may cause network congestion and load imbalance when scheduling the elephant flows in data center networks. To deal with these isuses, this paper proposed scheme ACO-SDN, which is based on ACO (ant colony) algorithm in SDN(Software Defined Network) data center networks. It built an ILP（Integral Linear Programing）model according to the flow scheduling problem, with the object of minimize the maximum link utilization. Then, it employed ant colony algorithm to solve the ILP model through redefining its parameters and operations. The optimal route was obtained for of rerouting elephant flows. The experimental results show that, in comparisons with ECMP(Equal-Cost multi-path Routing) and GFF (Global First Fit) algorithms, ACO-SDN algorithm lowers down the maximum link utilization rate, and effectively improves the network bisection bandwidth of networks.

Key words: data center network; SDN; elephant flow scheduling; ACO algorithm

0　引言

随着云计算、物联网和大数据技术的快速发展，数据中心已经成为现代计算基础设施的基石。数据中心内的通信量呈现以指数级增长的态势，对数据中心网络带宽的需求日益增加[1]。传统数据中心网络的树型结构存在着带宽受限、扩展性差等问题，无法满足数据中心内部通信量对网络带宽的需求[2]。为此，许多新型的数据中心网络体系结构[3~6]相继被提出，如 Fat-Tree[3]、Monsoon[4]、BCube[5]、Helios[6]等。与传统数据中心网络架构相比，Fat-tree 拓扑结构相对比较简单，提供多条等价路径和更高的对分带宽[7]，便于实现网络负载均衡。

当今数据中心网络广泛利用等价多路径（ECMP）算法[8]进行流量调度。当到达一个目的节点有多条可用的等价路径时，ECMP 对于到达同一目的地的多个数据流量，采用静态哈希散列将它们调度到多条等价路径上。由此 ECMP 可以实现网络的负载均衡以及数据流的快速转发。但是研究表明,数据中心内部的通信流量可以分为大象流和老鼠流[9, 10]，其中，大象流是指传输的数据量超过链路带宽 10%的数据流。通过对数据中心评测，文献[9，10]发现，90%的数据流属于老鼠流，但是大象流传输了超过 90%的字节，绝大部分数据中心流量被携带在很小一部分数据流中。在实际应用中，这两种数据流的传输性能要求不同，大象流对网络吞吐量要求较高，而老鼠流对时延要求较高[11]。文献[2]研究表明，ECMP 算法可以有效地调度大量的老鼠流，然而对于持续时间较长的大象流，ECMP 可能会将多

收稿日期：2018-07-13；**修回日期：**2018-08-31　　**基金项目：**国家自然科学基金资助项目（61363016）；国家教育部留学回国人员科研启动基金项目（[2014]1685）；内蒙古自治区自然科学基金资助项目（2015MS0605，2015MS0626）

作者简介：李宏慧（1970-），女，内蒙古赤峰人，教授，博士，主要研究方向为计算机网络、光网络、组合优化算法（lihh_chf@163.com）；杨光（1992-），男，硕士研究生，辽宁大连人，主要研究方向为软件定义网络；路海亮（1992-），男，内蒙古通辽人，硕士研究生，主要研究方向为软件定义网络；付学良（1966-），男，内蒙古赤峰人，教授，博导，博士，主要研究方向为智能计算、数据挖掘、农业信息化；申志军（1976-），男，河南信阳人，副教授，博士，主要研究方向为网络通信技术.

条大象流调度到同一条链路上，造成数据流碰撞、网络拥塞，使得网络负载不均衡，降低了数据中心网络的吞吐量。

作为一种新型的可编程网络技术，软件定义网络（SDN）[12]将控制平面从传统的网络设备中分离出来，通过在控制平面对流量的全局集中管控，可以方便灵活地调度网络流量和优化资源管理，为解决数据中心网络问题提供了新的机遇。

针对数据中心网络中 ECMP 调度大象流易于造成网络拥塞和负载不均衡等问题，在 Fat-Tree 网络架构下，本文提出了一种基于蚁群算法的 SDN 数据中心网络动态的大象流调度机制（ACO-SDN）。首先建立流量调度问题优化数学模型（ILP），在满足所有大象流调度要求的前提下，优化目标为最小化网络的最大链路利用率；然后通过对蚁群算法参数和操作重定义来求解 ILP，得到大象流的优化调度方案。ACO-SDN 根据检测到大象流和网络链路利用率，从数据中心网络全局的角度对大象流进行重路由。仿真实验结果表明，与 ECMP 和 GFF[2]算法相比，ACO-SDN 提高了数据中心网络的对分带宽，降低了最大链路利用率，实现了网络负载均衡。

1 相关工作

随着 SDN 技术的兴起，许多学者尝试利用 SDN 集中控制的方法对数据中心的流量进行管理，通过使用控制器实现对流量的实时调度。

文献[13]提出了一种动态的流量调度机制 Hedera 来最大化数据中心网络对分带宽，提出了全局首次适应（GFF）算法进行流量调度。对每一条大象流，GFF 算法遍历所有可能的路径，根据路径上链路的剩余带宽找到第一条满足流带宽需要的路径。

但是文献[13]利用交换机监测大象流，造成交换机的额外开销比较大。针对该问题，文献[2]提出了低开销、有效的流量调度算法 Mahout。该算法利用终端主机进行流量监测，一旦检测到大象流就会给控制器发信号，由控制器为大象流计算路径。

文献[14]研究了 SDN 数据中心的动态负载均衡调度（DLBS）问题。首先建立 DLBS 问题的数学优化模型，目标是在保证动态负责均衡的条件下最大化网络吞吐量；然后提出了高效的启发式算法求解优化模型，实现各时间槽内的网络负载均衡。

文献[15]提出了无须大象流检测的多路径路由方案，使得数据流完成时间最小化。该方案基于超时阈值的数据流会被移除的特性，将大象流分成若干老鼠流，并路由到所有可能的路径上，由此实现数据中心网络的流量多路径路由。

文献[16]针对 SDN 数据中心网络的大象流调度问题，利用稳定匹配理论对给问题建模并求解该模型。目标是在避免网络拥塞的同时使得网络性能最优。文中提出了新的基于稳定匹配的大象流调度算法实现数据中心网络的高可用性和低时延。

文献[17]提出了一种 SDN 大数据中心网络的差异化地调度算法，依据检测到的数据流类别，利用加权多路径调度算法和基于封锁岛路径设置算法分别调度老鼠流和大象流。并设计了一种算法，根据当前链路利用率动态的重调度数据流来实现负载均衡。目标是获得高吞吐量、低延时和负载均衡。

针对 Hedera 中的模拟退火算法未考虑当前网络链路带宽资源引起的流冲突等问题，文献[18]提出了基于模拟退火遗传算法的按需自适应(SAGA-AO) 流量调度机制。该机制首先筛选出网络中需要调度的流；然后利用模拟退火遗传算法(SAGA)，根据当前链路带宽资源状况进行全局搜索,得到流调度路径，由此提高了数据中心网络的平均对分带宽。

针对现有的流量调度算法可能造成链路负载不均衡和核心交换机冲突加剧的问题，文献[19] 提出了基于离散粒子群的流调度算法 DPSOFS。该算法从全局角度进行流量调度，可以有效快速地减少流冲突，提高网络对分带宽。

文献[20]提出一种可扩展的基于 SDN 的数据中心网络分段路由流调度策略来解决大规模 SDN 数据中心网络流量路由机制的可扩展性低引起的性能瓶颈问题。利用边缘交换机检查大象流，同时为满足其不同业务的 QoS 保证和网络可扩展性的要求，提出了在线最先适应算法。仿真结果表明该机制在降低了控制器总开销的同时提高了网络吞吐率。

文献[21]针对数据中心网络中大象流携带大量数据造成网络拥塞和负载不均衡的问题，提出基于 SDN 的大象流负载均衡机制 EFLB。该机制以轮询方式监听网络，当负载超过阈值时，控制器将检测到的大象流拆分为多个老鼠流，并计算出负载最小的下一跳交换机，确保负载均衡。该机制提高网络吞吐量，降低了网络时延，更好地实现网络负载均衡。

2 蚁群流量调度算法

本章首先介绍基于蚁群算法的流量调度机制 ACO-SDN，包括大象流检测和网络状态收集方法；然后建立流量调度问题的优化数学模型 ILP；最后给出用于求解 ILP 模型的基于蚁群算法的流量调度方法。

2.1 大象流流量调度机制 ACO-SDN

本文提出的流量调度机制流程如图 1 所示。具体流量调度流程如下：

a)当数据中心网络收到来自主机的数据流以后，首先按 ECMP 算法来调度数据流。同时，交换机中的 sflow 代理采集网络状态信息，包括链路使用情况以及数据流信息，每隔 2 s 将采集到的网络信息传输到控制器中的 sflow 收集器。

b)依据收到的信息，sflow 收集器计算链路利用率以及识别大象流。如果一条数据流的带宽超过了链路容量的 10%，则被标记为大象流。

c)sflow 收集器将大象流和链路利用率传给 floodlight 控制器。

d)控制器判断识别出的大象流途经链路利用率是否大于阈值 60%。如果是，转步骤 e)，否则转步骤 a)。

e)控制器根据当前链路使用状态，调用蚁群算法求解流量

调度数学模型，为大象流计算出一条最优路径。将该路径转换为流表项下发给交换机。

f)交换机根据新的流表项重路由拥塞链路上的大象流。转步骤 a)。

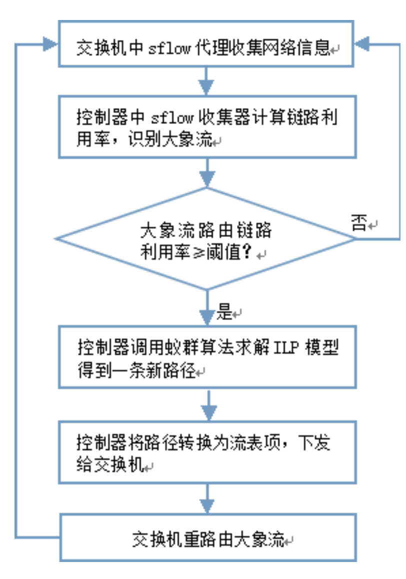


图 1　大象流检测及调度流程

2.2　流量调度问题数学模型

将数据中心网络拓扑表示为图 $G(V,L)$，其中：$V=\{v_1,v_2,\dots,v_n\}$ 表示网络中所有节点的集合；L 表示网络中所有链路的结合，L_i 为节点 v_i 的所有数据流流入链路集合，L_i' 为节点 v_i 的所有数据流流出链路集合，链路 $l\in L$ 的容量为 C_l，链路利用率为 u_l。设当前网络需要调度的大象流集合为 F，大象流 $f\in F$ 的带宽为 b_f，源点和终点分别为 v_s^f 和 v_t^f。引入变量 $x_l^f\in\{0,1\}$ 表示大象流 f 的路由是否经过链路 l。

链路利用率是指流经该链路的所有大象流带宽之和与链路容量的比值，如式（1）所示。

$$u_l=\frac{\sum_{f\in F}x_l^f b_f}{C_l} \tag{1}$$

最大链路利用率 $u^{\max}$ 是指网络中所有链路利用率的最大值，即

$$u^{\max}=\max\{u_l\}\qquad l\in L \tag{2}$$

流量调度问题的数学模型如下所示：

$$\min\quad u^{\max} \tag{3}$$

s.t.

$$\sum_{f\in F}x_l^f b_f\le C_l\qquad l\in L \tag{4}$$

$$\sum_{l\in L_i}x_l^f b_f-\sum_{l\in L_i'}x_l^f b_f=\begin{cases}-b_f & v_i=v_s^f,\ f\in F\\ b_f & v_i=v_d^f,\ f\in F\\ 0 & v_i\in V-\{v_s^f,v_d^f\}\end{cases} \tag{5}$$

$$x_l^f\in\{0,1\}\qquad l\in L,\ f\in F \tag{6}$$

其中：式（3）为流量调度问题的优化目标，即最小化网络的最大链路利用率；式（4）～（6）为流量调度时应满足的约束条件。式（4）确保链路 l 承载的数据流带宽总和不会超过该链路的容量；式（5）为流守恒约束条件，确保数据流途经各中间结点时进入一个结点的流量等于离开该结点的流量；式（6）定义了变量的取值范围。

该数学模型属于整型线性规划（ILP）数学模型。为了得到大象流的路由调度方案，本文采用蚁群优化算法求解该模型。

2.3　蚁群优化算法

为了求解上述 ILP 模型，蚁群算法和遗传算法都是强有力的工具。这两种算法的区别在于，蚁群算法利用信息素的积累及更新，最终得到最优解；而遗传算法具有快速进行全局解空间搜索能力，但由于没有利用反馈信息，导致冗余的迭代，求解效率较低[23]。

蚁群算法是由 Marco Dorigo 首先提出来的寻找最优路径的近似优化算法，其灵感来源于蚁群在觅食过程中发现最短路径的行为[24]。蚂蚁在其途经的路径上会释放信息素进行信息传递，蚁群内的其他蚂蚁会感知到信息素，并且会沿着信息素浓度较高路径行走，每只路过的蚂蚁又会释放信息素，从而形成了一种正反馈机制。经过一段时间以后，整个蚁群就会沿着最优路径到达食物源了。

利用蚁群算法求解上述 ILP 模型的基本思路就是，用蚂蚁觅食的行走路径表示大象流的可行路由解，整个蚂蚁群体的所有路径构成大象流可选路由的解空间。路径较短的蚂蚁释放的信息素量较多，随着时间的推进，较短的路径上累积的信息素浓度逐渐增高，选择该路径的蚂蚁个数也愈来愈多。最终，整个蚂蚁会在正反馈的作用下集中到最佳的路径上，此时对应的便是大象流路由的最优解。具体算法步骤如下所述：

a)初始化参数。参数分为网络状态参数，大象流参数和蚁群算法参数。其中：

(a)网络状态参数包括节点集合 V、链路集合 L、链路 $l\in L$ 的容量 C_l 和链路利用率为 u_l。

(b)大象流参数包括需要调度的大象流 f 的带宽 b_f、源点 v_s^f 和终点 v_t^f、流的持续时间。

(c)蚁群算法参数包括蚂蚁个数 m、蚂蚁的初始位置 V、网络信息素总量 Q、信息素启发式因子 α、期望启发因子 β、信息素挥发系数 ρ、网络中各链路的初始信息素 τ、蚂蚁途经的最多节点数 $N_{\max}$ 和算法迭代次数 I、蚂蚁 k 的禁忌链路表 TB_k。

b)用 $\tau_{ij}(t)$ 表示 t 时间从节点 i 到 j 路径的信息素浓度，用 $\eta_{ij}(t)$ 表示 t 时间从节点 i 到 j 的启发式信息，即两节点间链路利用率的倒数，$\eta_{ij}(t)=1/u_l$。在遍历所有可行路径过程中，每只蚂蚁 k 依据 $\tau_{ij}(t)$ 和 $\eta_{ij}(t)$ 按概率 $p_{ij}^k(t)$ 选择下一个节点 j，将蚂蚁 k 经过的链路保存到禁忌链路表 TB_k。其中：

$$p_{ij}^k(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^\alpha[\eta_{ij}(t)]^\beta}{\sum_{n\in V_k(i)}[\tau_{in}(t)]^\alpha[\eta_{in}(t)]^\beta}, & j\in V_k(i)\\ 0, & \text{其他}\end{cases}$$

$V_k(i)$ 表示蚂蚁 k 下一步允许访问的节点集合，$V_k(i)=V-TB_k$。

c)重复步骤 b)，当所有蚂蚁途经的节点数达到 N_{max} 后，停止本次迭代，迭代次数加 1。

d)对于成功到达目的地的蚂蚁 k，依据其 TB_k 更新路径上的信息素浓度，更新公式为 $\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho\Delta\tau_{ij}(t)$。其中 $\Delta\tau_{ij}(t)$ 为路径上的信息素增量。对于蚂蚁 k 没有经过的路径，$\Delta\tau_{ij}(t)$ 值为 0，否则 $\Delta\tau_{ij}(t)=Q/L_k$，L_k 为第 k 只蚂蚁在本次所走的路径长度。

e)完成规定的最大迭代次数 I 后停止路径搜索，从所有到达目的地的蚂蚁禁忌表中按优化目标选出最优路径。

为了尽可能地避免蚁群算法陷入局部最优，本文采取以下两个措施：

a）差异化地对网络链路的信息素浓度进行初始化。将待调度大象流源点与终点之间的 K-最短路径作为备选路径，对每条备选路径，将其途经链路的初始信息素浓度设置为信息素总量除以路径长度；而其他链路信息素浓度设置为 0。

b）采用轮盘赌算法，增加蚁群算法的全局搜索能力，进一步优化解的质量[25]。

3 实验结果与分析

为了验证 ACO-SDN 优化算法的性能，本文利用 Floodlight 控制器和 Mininet[22]仿真平台搭建 Fat-tree 数据中心网络，采用平均对分带宽和最大链路利用率作为算法的性能评价指标。对分带宽指的是将一个网络中的所有主机分为对等两部分时，需要断开的最小链路数的带宽总和。对分带宽越大则网络的整体传输性能越强。通过与 GFF 算法和当前数据中心中普遍使用的 ECMP 算法进行比较，实验结果显示，ACO-SDN 在平均对分带宽、最拥塞链路利用率等性能指标方面呈现优势。

3.1 实验设置

1）网络拓扑结构

在 Mininet 仿真平台上，本文搭建了 k=4 的 Fat-tree 数据中心网络架构，如图 2 所示，所有的交换机均为 OpenFlow 交换机。各条链路的带宽均设定为 100 Mb/s。

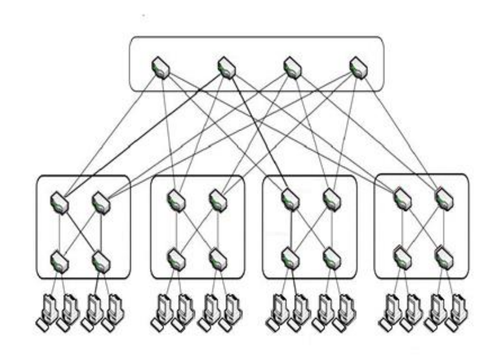


图 2 K=4 Fat-Tree 拓扑结构

2）网络负载

目前，由于隐私和安全问题，无公开发表的数据中心网络负载数据可用。最近的有关数据中心网络流量特征的研究[9, 26]表明，随着时空变化，通信量矩阵变化巨大。为了评估提出的大象流调度机制 ACO-SDN，本文依据文献[3，13，18，19]确定所采用的通信模式。

本实验使用 Iperf 工具产生数据流量。每台主机产生的流长度不同，服从指数分布；流产生的时间遵从泊松分布；流的目的主机根据某种通信模式来确定，本文采用如下三种通信模式：

a）间隔模式 Stride(*i*)。下标为 x 的主机向下标为（$x+i$）*mod n* 的主机发送数据，其中 n 为网络中主机数量。

b）交错模式 Staggered (*pEdge, pPod*)。每台主机以概率 *EdgeP* 向同属于一个边缘交换机的主机发送数据，以概率 *pPod* 向同属于一个 *Pod* 的主机发送数据，以概率 *1-EdgeP-pPod* 向其他 *pod* 内主机发送数据。

c）随机模式 Random。每台主机以相等概率随机向网络中其他的主机发送数据。

3）算法参数设置

如 2.3 节所述，蚁群优化算法的参数分为网络参数、大象流参数和蚁群算法参数。在进行模拟仿真实验时，这些参数的设置及依据如下：

a）网络参数。

网络状态参数中，节点集合 V 为如图 2 所示的 Fat-tree 数据中心网络的所有节点，包括交换机和服务器节点。链路集合 L 为该网络的所有链路。每个链路的容量均为 100 M[21]。链路利用率 u_l 依据链路的当时负载情况，按式（1）计算。

b）大象流参数。

大象流 f 的源点 v_s^f 和终点 v_t^f 均为服务器节点，根据采用的通信模式在生成大象流时进行设置。大象流定义为带宽超过链路容量 10%的数据流[2]，所以本文将大象流带宽 b_f 定义为 10 M～20 M 间的随机数。大象流的长度在大象流生成时设定。

c）蚁群算法参数。

蚁群算法参数的设置直接影响蚁群优化算法的性能。本文依据文献[27，28]的研究、流量调度问题的要求和多次仿真实验结果进行设置。

蚁群算法的主要参数设置及依据如表 1 所示。

表 1 蚁群算法主要参数值

参数	设定值	建议值[1,2]
蚂蚁个数 m	10	[0.6n, 0.9n]
信息素启发式因子 α	2	[1, 5]
期望启发因子 β	3	[1, 5]
信息素挥发系数 ρ	0.6	[0.5, 0.8]
信息素总量 Q	100	100

蚂蚁的初始位置 V 为待调度大象流的源点。蚂蚁 k 的禁忌链路表 TB_k 设置为空值。通过多次实验测试，迭代次数在 50～100 间时，算法性能稳定，所以本算法将迭代次数 I 设置为中间值 75。在进行仿真实验时，两个节点之间的路径长度限定为 11 跳。相信在 k=4 的 Fat-Tree 网络拓扑中，超过 11 跳的路径是最优解的可能性非常小，所以将蚂蚁途经的最多节点数，即交换机数设定为 10。各链路的初始信息素 τ 设置如第 2.3 节所

述。

3.2　平均对分带宽比较

为了比较 ECMP、GFF 和 ACO-SDN 三种算法的平均对分带宽，本文采用上述三种通信模式分别进行了多次实验，每次实验持续约 60 s，取中间部分测量对分带宽。

1）Stride 通信模式

在 Stride(*i*)间隔通信模式下，选取了 *i* = 4、*i* = 6 和 *i* = 8 三种情况进行实验。在 Stride(4)、Stride(6)和 Stride(8)三种间隔模式下，图 3 展示了 ECMP、GFF 和 ACO-SDN 三种算法的平均对分带宽比较结果。从图 3 可以发现，在这三种间隔通信模式下，ACO-SDN 算法平均对分带宽均高于 ECMP 和 GFF 算法。ACO-SDN 平均对分带宽与 ECMP 相比，提高了 19%~26%，与 GFF 相比提高了 8%~17%。

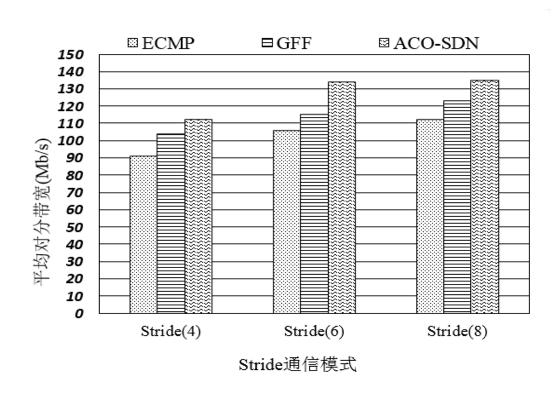


图 3　Stride 模式下平均对分带宽比较

2）Staggered 通信模式

在 Staggered 交错通信模式下，选取了 Staggered（0，0.2）和 Staggered（0.0.4）两种交错模式。每种交错模式选取两组实验。三种算法的平均对分带宽比较如图 4 所示。从图 4 可以看出，ACO-SDN 算法在这两种通信模式下均优于 ECMP 和 GFF 流量调度算法。与 ECMP 相比，ACO-SDN 算法平均对分带宽分别提高了 8%~21%，与 GFF 相比，提高了 5%~15%。

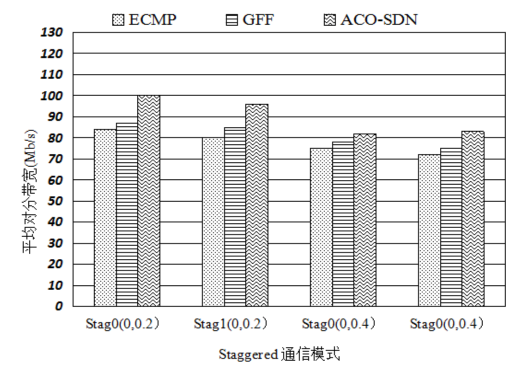


图 4　Staggered 模式下平均对分带宽比较

3）Random 通信模式

在 Random 随机通信模式下，选取了三组实验 Random1、Random2 和 Random3。三种算法的平均对分带宽比较结果如图 5 所示。从图 5 可以看出，ACO-SDN 算法在平均对分带宽上均优于 GFF 和 ECMP 流量调度算法。与 ECMP 和 GFF 相比，ACO-SDN 算法的平均对分带宽分别提高了~23%和~14%。

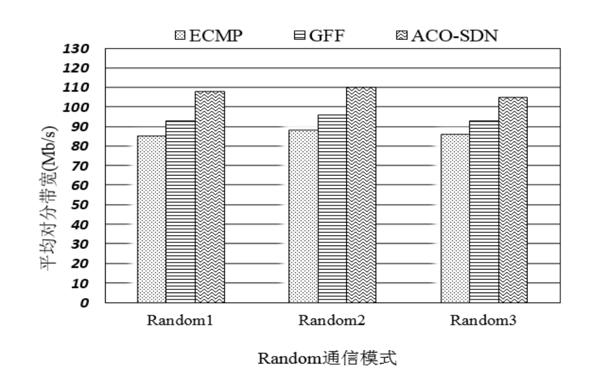


图 5　Random 模式下平均对分带宽比较

综上所述，在三种不同的通信模式下，ACO-SDN 算法的平均对分带宽均高于 ECMP 和 GFF 算法。由于 ECMP 随机将大象流调度到一条最短路径，未考虑当前链路状态，导致大象流冲突增多，所以在三种算法中平均对分带宽最低。GFF 算法对于每个到达的大象流，根据当前链路利用率选取第一条满足条件的路径调度大象流，减少了数据流的冲突，与 ECMP 相比提高了网络的平均对分带宽。ACO-SDN 算法根据当前链路利用率计算出全局最优的路径来调度大象流，从而减少了大象流的冲突，提高了网络平均对分带宽。

3.3　最大链路利用率比较

为了进一步验证算法性能，本文从最大链路利用率角度对 ACO-SDN、ECMP 和 GFF 算法进行了比较。最大链路利用率指的是全网范围内链路利用率的最大值，它反映了网络中链路带宽的使用情况。如果网络负载比较均衡，则数据流从源地址到目的地址的各条路径链路利用率应该比较均衡，才能充分利用网络中多路径的优势进行流量调度，减少数据流冲突，提高网络吞吐量。

实验选取了交错通信模式 Staggered（0，0.3），以第一个 Pod 中的主机为数据源，向其他 Pod 中的主机发送数据流，流量带宽固定为 20 M，持续时间为 60 s。采用这种实验的目的是为了造成网络负载不均衡和较高的链路负载，从而可以更好地检测算法的最大链路利用率。在实验中，利用 ECMP、GFF 和 ACO-SDN 流量调度算法分别进行多次实验，直到得到比较稳定的结果。

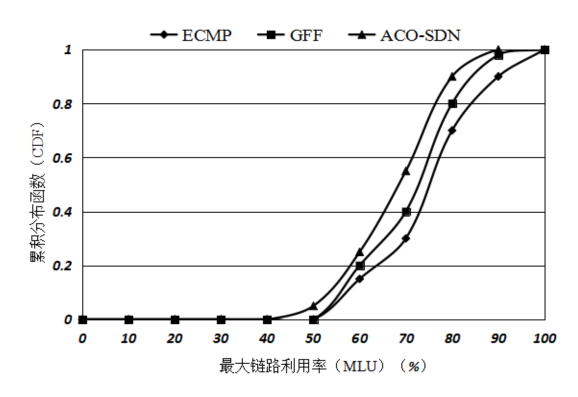


图 6　Staggered 模式下最大链路利用率比较

三种算法的最大链路利用率的累计分布函数如图 6 所示。图中横轴为最大链路利用率（MLU），纵轴为 MLU 的累计分

布函数（CDF），曲线表示不同算法下 MLU 的 CDF。从图中可以看出，ACO-SDN 算法最大链路利用率比 GFF 降低 5% 左右，比 ECMP 降低 8%左右。ACO-SDN 算法的 MLU 集中在 60%~70%间，而 ECMP 算法的 MLU 集中在 70%~80%间。这主要是由于 ACO-SDN 算法实时监测大象流路径上的链路，当发现某链路利用率超过阈值时，则启动 ACO-SDN 算法根据当前网络链路状态计算出一条新路径来重路由大象流，降低了最大链路利用率，减少了大象流冲突。而 ECMP 算法是静态流量算法，在流量调度时随机将大象流调度到一条最短路径上，但是没有考虑当前的链路状态，增加了大象流冲突的可能性和最大链路利用率，所以 ECMP 的性能在这三种算法中性能最低。GFF 算法在路由大象流时考虑了网络的链路利用率，所以性能比 ECMP 有所提高；但是它根据大象流到达的先后顺序，将它们路由到第一条满足条件的路径上，所以最大链路利用率介于 ACO-SDN 和 ECMP 算法。

4 结束语

针对传统的流量调度方法易于造成大象流冲突，进而导致网络拥塞和性能下降的问题，本文提出了基于 SDN 的 Fat-Tree 数据中心网络大象流调度机制。首先对大象流调度问题建立整型线性规划(ILP)优化模型，并利用蚁群优化算法求解该 ILP 模型，得到大象流的近似最优调度方案。通过 Floodlight 控制器和 Mininet 模拟平台进行了仿真实验。实验结果表明，本文提出的 SDN 数据中心网络大象流调度机制 ACO-SDN 优于 ECMP 和 GFF 算法，在 Stride、Staggered 和 Random 三种通信模式下，与 ECMP 和 GFF 算法相比，ACO-SDN 较大地提升了网络平均对分带宽，同时降低了最大链路利用率。

参考文献：

[1] 蔡岳平，王昌平. 软件定义数据中心网络混合路由机制 [J]. 通信学报, 2016, 37 (4): 44-52. (Cai Yueping, Wang Changping. Software defined data center network with hybrid routing [J]. Journal on Communications, 2016, 37 (4): 44-52.)

[2] Curtis A R, Kim W, Yalagandula P. Mahout: low-overhead datacenter traffic management using end-host-based elephant detection [C]// Proc of IEEE INFOCOM. 2011: 1629-1637.

[3] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture [C]// Proc of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications. 2008: 63-74.

[4] Greenberg A, Lahiri P, Maltz D A, *et al.* Towards a next generation data center architecture: scalability and commoditization [C]// Proc of ACM Workshop on Programmable Routers for Extensible Services of Tomorrow. [S. l.] : ACM Press, 2008: 57-62.

[5] Guo Chuanxiong, Lu Guohan, Li Dan, *et al.* BCube: a high performance, server-centric network architecture for modular data centers [J]. ACM SIGCOMM Computer Communication Review, 2009, 39 (4): 63-74.

[6] Farrington N, Porter G, Radhakrishnan S *et al.* Helios: a hybrid electrical//optical switch architecture for modular data centers [J]. ACM SIGCOMM Computer Communication Review, 2010, 40 (4): 339-350.

[7] Escudero-Sahuquillo J, Garcia P J, Quiles F J, *et al.* A new proposal to deal with congestion in InfiniBand-based fat-trees [J]. Journal of Parallel & Distributed Computing, 2014, 74 (1): 1802-1819.

[8] Hopps C. Analysis of an equal-cost multi-path algorithm [S]. RFC 2992, 2000.

[9] Kandula S, Sengupta S, Greenberg A, *et al.* The nature of data center traffic: measurements & analysis [C]// Proc of ACM SIGCOMM Conference on Internet Measurement. 2009: 202-208.

[10] Benson T, Akella A, Maltz D A. Network traffic characteristics of data centers in the wild [C]// Proc of the 10th ACM SIGCOMM Conference on Internet Measurement. 2010: 267-280.

[11] Li Dan, Xu Mingwei, Zhao Hongze, *et al.* Building mega data center from heterogeneous containers [C]// Proc of IEEE International Conference on Network Protocols. [S. l.] : IEEE Press, 2011: 256-265.

[12] Mckeown N, Anderson T, Balakrishnan H, *et al.* OpenFlow: enabling innovation in campus networks [J]. ACM SIGCOMM Computer Communication Review, 2008, 38 (2): 69-74.

[13] Al-Fares M, Radhakrishnan S, Raghavan B, *et al.* Hedera: dynamic flow scheduling for data center networks [C]// Proc of Usenix Symposium on Networked Systems Design and Implementation. 2010: 281-296.

[14] Tang Feilong, Yang L T, Tang Can, *et al.* A dynamical and load-balanced flow scheduling approach for big data centers in clouds [J]. IEEE Trans on Cloud Computing, 2016, 99 (1) , 1-14.

[15] Chakraborty Suchandra, Chen Chien. A low-latency multipath routing without elephant flow detection for data centers [C]// Proc of IEEE International Conference on High Performance Switching and Routing. Yokohama: IEEE Press, 2016: 49-54.

[16] Zhang Yuxiang, Cui Lin, Zhang Yuan. A stable matching based elephant flow scheduling algorithm in data center networks [J]. Computer Networks, 2017, 120 (2017): 186-197.

[17] Zhang Heteng, Tang Feilong, Barolli L. Efficient flow detection and scheduling for SDN-based big data centers [J/OL]. Journal of Ambient Intelligence & Humanized Computing, 2018, https://doi. org/10. 1007/s12652-018-0783-6.

[18] 王文涛，郑芳，王玲霞，等. 基于SDN的数据中心网络流量调度机制的设计与实现 [J]. 中南民族大学学报：自然科学版，2016, 35 (3): 135-140. (Wang Wentao, Zheng Fang, Wang Lingxia, *et al.* Design and implementation of flow scheduling mechanism based on SDN for data center network [J]. Journal of South-Central University for Nationalities: Nat. Sci. Edition, 2016, 35 (3): 135-140.)

[19] 林智华，高文，吴春明，等. 基于离散粒子群算法的数据中心网络流量调度研究 [J]. 电子学报, 2016, 44 (9): 2197-2202. (Lin Zhihua, Gao Wen,

Wu Chunming, Li Yongyan. Data center network flow scheduling based on DPSO algorithm [J] , Acta Electronica Sinica, 2016, 44 (9): 2197-2202.)

[20] 伊鹏，刘洪，胡宇翔．一种可扩展的软件定义数据中心网络流调度策略 [J]. 电子与信息学报，2017, 39 (4): 825-831. (Yi Peng, Liu Hong, Hu Yuxiang. A scalable traffic scheduling policy for software defined data center network [J]. Journal of Electronics & Information Technology, 2017, 39 (4): 825-831.)

[21] 金玲，束永安．数据中心网络中基于 SDN 的大象流负载均衡的研究 [J/OL]. 计算机应用研究，2019 (1): 1-5. [2018-05-30]. http://kns. cnki. net/kcms/detail/51. 1196. TP. 20180208. 1714. 094. html. (Jin Ling, Shu Yongan. Research on load balancing of elephant flow based on SDN in data center network [J]. Application Research of Computers, 2019 (1): 1-5. [2018-05-30]. http://kns. cnki. net/kcms/detail/51. 1196. TP. 20180208. 1714. 094. html.)

[22] Lantz B, Heller B, Mckeown N. A network in a laptop: rapid prototyping for software-defined networks [C]// Proc of ACM Workshop on Hot Topics in Networks. 2010: 1-6.

[23] 康岚兰，李康顺．蚁群算法在求解 TSP 问题上与遗传算法的对比研究 [J]. 计算机系统应用, 2008, 17 (10): 60-63. (Kang Fenglan, Li Kangshun. A comparison study of GA and ACA on TSP [J]. Computer Systems & Applications, 2008, 17 (10): 60-63.)

[24] 段海滨，张祥银，徐春芳．仿生智能计算 [M]. 北京：科学出版社 2011. (Duan Haipin, Zhang Xiangyin, Xu Chunfang. Bio-inspired computing [M]. Beijing: Science Press, 2011.)

[25] 马振．改进蚁群算法及其在 TSP 中的应用研究 [D]. 青青岛：岛理工大学, 2016. (Ma Zhen. Research on the improvement of ant colony algorithm and its application in TSP [D]. Qingdao: Qingdao University of Technology, 2016.)

[26] Greenberg A, Hamilton J R, Jain N, *et al.* VL2: a scalable and flexible data center network [J]. Communications of the Acm, 2009, 54 (4): 95-104.

[27] 詹士昌，徐婕，吴俊．蚁群算法中有关算法参数的最优选择 [J]. 科技通报, 2003, 19 (5): 381-386. (Zhan Shichang, Xu Jie, Wu Jun. The optimal selection on the parameters of the ant colony algorithm [J]. Bulletin of Science and Technology, 2003, 19 (5): 381-386.)

[28] 徐红梅，陈义保，刘加光，等．蚁群算法中参数设置的研究 [J]. 山东理工大学学报：自然科学版，2008, 22 (1): 7-11. (Xu Hongmei, Chen Yibao, Liu Jiaguang, *et al.* The research on the parameters of the ant colony algorithm [J]. Journal of Shandong University of Technology: Natural Science Edition, 2008, 22 (1): 7-11.)